EFFECT OF FEA EPISTEMIC UNCERTAINTY ON DESIGN OF CELLULAR METAMATERIALS WITH NON-LINEAR MECHANICAL BEHAVIOR

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

M.S.

by

Zelin Linghu

May 2018

ABSTRACT

Finite element analysis (FEA) is widely used in design of metamaterials to predict the mechanical behavior as a function of design variables such as geometry and material choice. While FEA modeling is mature, there are numerous modeling decisions that influence the reliability and outcomes of the simulation, especially for large material deformation and highly non-linear behaviors such as plasticity and buckling. These decisions become sources of epistemic uncertainty that propagate onto simulation outcomes and thus potentially affect designs. This thesis investigates the impact of two uncertainty sources that are especially relevant in studying buckling – geometric imperfections and frictional interactions – on a honeycomb-based cellular material. FEA package Abaqus is selected here and Python scripting is developed to achieve the FEA implementation and design automation. Uncertainty is characterized and quantified based on a large computational experiment, and the impact of the uncertainty on a set of simple robust design optimization and Bayesian optimization problems is illustrated. The results show that, combined, both factors introduce significant variability (up to 20%) in the mechanical response of the material, and consequently lead to changes (up to 10%) of the optimum design in a robust design optimization framework. This study suggests that ignoring these sources of uncertainty may lead to suboptimal designs for this class of material design problems. The methodology and codes developed in this thesis can be used to further explore the mechanical properties of metamaterials and facilitate material design.

BIOGRAPHICAL SKETCH

Zelin Linghu is pursuing a Master of Science degree in the department of mechanical engineering at Cornell University. He received a B.S.in automotive engineering at Tsinghua University in Beijing, China. His research interest is focused on applied mechanics and design optimization methodology.

This thesis is dedicated to my family.

ACKNOWLEDGEMENTS

I would like express my sincere gratitude to my committee members, Professor Silberstein and Professor Selva. They gave me valuable guidance and encouragement in my research. This thesis cannot be done without their efforts. Also great thanks to my labmates, especially Naigeng and Suwon, for their assistance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Polymer-based metamaterials are compelling because of the potential to achieve properties that are difficult to achieve through traditional intrinsic material design, such as negative Poisson's ratio, bi-stability, large energy absorption, low density, high anisotropy, large tension/compression asymmetry, and fluid-like shear to dilatational response ratios [1, 2, 3, 4, 5]. With the proliferation of 3D printing, the accessible design space for mechanical metamaterials has increased significantly, since both shape and dimensions can be widely varied. It is promising to create materials with target mechanical properties by prescribing both intrinsic material properties and microcellular architecture by 3D printing. While finite element analysis (FEA) is widely adopted to predict the mechanical responses of the materials, exploring the large design space is especially challenging when studying large deformation properties such as strength, toughness, and failure strain, since those are substantially more expensive to evaluate than elastic properties [6].

1.2 Crashworthiness Design of Honeycombs

This thesis focuses on cellular materials and specifically honeycombs, basically an array consisting of repeated hexagons. This class of materials is interesting because their crushing behavior is unstable and nonlinear in nature. More-

over, honeycombs are widely used in traditional macro-scale metallic material-based crashworthiness design problems. The major application of honeycombs is serving as the energy absorption structures of vehicles during collision. Designers prefer structures that are light weight and have a high capability of protection, which are also the desired properties in the design of mechanical metamaterials. Commonly used metrics for traditional structural crashworthiness design include injury-based metrics such as maximum acceleration and peak force, and energy-based metrics like absolute energy absorption, specific energy absorption (SEA), and mean crushing force [7]. These metrics describe the mechanical performance of the structure/material. In addition to the theoretical [8, 9, 10] and experimental [11, 12, 13] studies, honeycombs are extensively studied by numerical simulations, i.e., the finite element method, especially in the context of engineering design and optimization [14, 15, 16]. However, few studies use honeycomb as the base structure to design metamaterials and investigate their responses during an unstable deformation process.

1.3 FEA of Mechanical Metamaterials

A common and fairly general approach to determine the mechanical properties of a mechanical metamaterial is to conduct finite element simulations of the repeat unit of the cellular geometry known as the representative volume element (RVE) [17, 18]. The RVE is subjected to periodic boundary conditions in order to mimic the boundary conditions that each cell will experience while surrounded by identical cells. While FEA is a fairly mature and standardized methodology, available to non-mechanics experts through a variety of software programs, there are numerous expert-level decisions a modeler must make that

have an ill-quantified effect on the simulated result. Examples of such decisions are: Is hard contact a good enough approximation of how a material interacts with itself? Is more than one geometric repeat needed in the RVE to capture pertinent cooperative deformation modes? How many buckling modes need to be seeded and what should their relative magnitude be? Is the intrinsic time dependence of the material properties important? Typically, these choices matter more for large deformation and highly non-linear problems than for linear elastic, small deformation problems. For some of these questions, the designer could simply include as complex relations as possible, however this can lead to vast increases in computational cost. The answer to other questions is often unknown due to the absence of readily available and relevant experimental data, so the designer is forced to make choices that are only eventually validated in part by the agreement (or disagreement) of the realized metamaterial mechanical properties with those predicted.

1.4 Surrogate Modeling

FEA simulations for large deformation properties are often too computationally expensive to use directly in a design context, where a large number of function evaluations may be needed to identify a globally or at least locally optimal design. For this reason, statistical models that approximate the FEA simulation are often used instead. These statistical models are called surrogate models or metamodels in the literature.

Surrogate modeling can be seen as a statistical learning problem in which an unknown function $F : X \rightarrow Y$ is to be approximated based on a number of

example pairs (x_i, y_i), $i = 1, \ldots N$. Most often, the function to be approximated is either the mapping between the design space X and the objective space Y, or the error between that mapping and the one predicted by the original FEA simulation. In the vast majority of cases, the objective space consists of continuous variables $Y \subseteq \mathbb{R}^N$, and thus a regression formulation is used. This is the formulation adopted here. Classification formulations in which the continuous objective space Y is segmented into a number of discrete regions and the goal is to predict the region in which a given point $x \in X$ falls have also been explored to some extent (e.g., [19]), but are left out of the scope of this study.

A variety of algorithms have been used for regression-based surrogate modeling [20]. The most straightforward approach is to use design of experiments techniques such as fractional designs, Box-Benken designs, orthogonal arrays or Latin hypercubes [21] to define the list of examples, then run the expensive simulation at those points, and use polynomial regression techniques to determine a response surface. Then, this response surface calculated offline is used in the optimization in place of the original simulation. Many other algorithms have been used for regression, including Radial basis functions [22], neural networks [23], decision trees [24], kriging [25], Bayesian optimization [26], or ensemble methods [27, 28, 29] among others. In this study, response surface method and Bayesian optimization method are used to illustrate the importance of FEA epistemic uncertainty.

It is important to note that regression methods often rely on strong assumptions about the prior parameter distributions in the simulations, so it is important to check whether those assumptions are actually fulfilled in the problem at hand. For example, Sacks et al. argued that some fundamental assumptions

behind least squares and other similar statistical learning techniques, such as the fact that the errors are independent and identically distributed and follow zero-mean white Gaussian distributions, are unfounded for engineering simulations, since they are deterministic computational experiments. They proposed kriging as a better approach, in which the function is assumed to be a stochastic process that can be modeled by a multi-variate Gaussian distribution with a certain mean vector and non-zero covariance matrix [25]. More generally, Simpson et al. [20] provides some guidelines to choose a surrogate modeling technique for deterministic engineering simulations.

Sequential or adaptive methods have also been developed, in which instead of computing the surrogate model offline, the surrogate model is computed during the optimization. This allows allocating more fidelity to regions that are more interesting, either because uncertainty is high, or because they have promise in terms of the objective function. Bayesian optimization is a prominent example of this popular online approach [26].

1.5 Modeling Uncertainty

Surrogate modeling FEA simulations requires characterizing and quantifying the uncertainty present in these simulations. Generally, the literature distinguishes between aleatory uncertainty due to inherent randomness in physical phenomena and their measurements, and epistemic uncertainty due to our lack of knowledge of those physical phenomena that leads to systematic biases in our models. We note that FEA simulations are deterministic computational experiments, and thus in this case there is no aleatory uncertainty, just epistemic

uncertainty.

Epistemic uncertainty in FEA simulations comes from the fact that the modeler may be unsure about what values to choose for various model parameters, including the choice of meshing, solver, interaction model, and boundary conditions among others. The importance of characterizing model (epistemic) uncertainty has been recently emphasized in the literature. For example, Zhang et al. [30] developed a methodology to segment the design space into physically meaningful regions based on the level of uncertainty in the surrogate model. While probability theory is the most widely used approach to modeling uncertainty in engineering design, researchers have also attempted to model epistemic uncertainty using other theories, such as imprecise probabilities [31, 32], fuzzy sets [33, 34], or Dempster-Shafer theory [35, 36]. In this study, we start from a probabilistic framework, since it is the most common approach in robust design.

1.6 Robust design

Robustness in engineering design can be loosely defined as the ability of a product or system to satisfy stakeholder needs or requirements in the presence of various sources of uncertainty in design variables (control factors), parameters (noise factors), or objectives (response factors) without removing them [37, 38]. The robust design literature emphasizes sensitivity in performance, whereas the reliability-based design literature emphasizes sensitivity of the satisfaction of constraints. This study focuses on the former. "Non-robust" approaches to design find the optimal design in the absence of uncertainty, or assuming an

"average" or "most likely" case or scenario. However, uncertainty may change the performance of these optimal designs, making them suboptimal. Indeed, as noted in Shumacher and Olsinka [39] for the specific case of structural design, small changes from the expected values may result in important changes in structural behavior and thus performance. Thus, the optimal design when considering uncertainty will in general be different from the deterministic optimal design (see for example the classical example of the parking garage by De Neufville et al. [40]). In fact, some authors have noted that optimal designs in the deterministic sense are often the least robust to variations [41].

Robust design methodologies seek to identify designs that instead of leading to optimal performance for the expected case, lead to acceptable performance on a wide range of cases. The specific interpretation of this definition leads to different mathematical formulations. Perhaps the most widely known robust design methodology is the Taguchi method [37, 42], which aims to find the design that is the least sensitive to variations while meeting all design requirements. Chen et al. [43] proposed a method integrating the response surface methodology described above with compromise decision support problems, a variation of goal programming [44] in which the user sets target values for a number of design attributes, and the objective is to minimize the cumulative (or worst case) deviation with respect to those target values [45]. Many other robust design methodologies exist in the literature [46, 47], such as stochastic expansions [48], Suh's axiomatic design theory [49], and perturbation theory [50]. Some methods from finance are also applicable, such as mean-variance optimization [51], variations of Value-At-Risk [52], and extreme value theory [53]. The formulation used in this study to illustrate the importance of FEA epistemic is similar in spirit to Chen's method, combining a goal programming approach with a

response surface and a minimax formulation.

1.7 Research Goals, Approach, and Contributions

The goal of this thesis is to characterize FEA epistemic uncertainty and quantify its impact on robust design for a specific class of materials, namely honeycomb-type cellular materials. This class of problems is interesting because the crushing behavior is unstable and nonlinear in nature. Moreover, honeycombs are widely used in traditional macro-scale metallic material based crashworthiness design problems. FEA simulations are conducted using Abaqus to predict mechanical responses of the honeycombs under large axial compression. Two sources of uncertainty are studied: frictional interaction and initial imperfection. A computational experiment based on a 6×6 factorial design is used to characterize the uncertainty. A design problem based on a goal programming formulation with a single design variable (wall thickness) and target values for three material properties (effective stiffness, transition force, and specific energy absorption) is used to illustrate the impact of uncertainty on the optimal design. The results show that friction and imperfection could lead to a variability of up to 20% in terms of mechanical performance of the material, and further, change the optimum design choice by up to 10%. Moreover, it is found that the normality assumption in some surrogate models may be violated.

CHAPTER 2

FEA SIMULATIONS AND UNCERTAINTY QUANTIFICATION

2.1 Geometry and Periodic Boundary Conditions

To simulate the bulk properties of a honeycomb array, periodic boundary conditions (PBC) are implemented for the x and y-directions of the representative volume element (RVE), as shown in Figure 2.1. The red dashed box in Figure 2.1 b) illustrates how the RVE is selected out of a honeycomb array. The height of the RVE is 40 mm and the wall width of a cell is 10 mm. The wall thickness, denoted by t in Figure 2.1 c), is the single design variable in this investigation. As noted in the introduction, PBCs make the simulated cell act as if it is surrounded by identical cells. For the theoretical foundation of PBC, refer to [18, 54]. The core idea is that after generating the mesh, the user ties the corresponding nodes on the opposite faces of RVE together by adding constraints. In Abaqus, the user first gets the coordinates of the node from the *.input* file generated by Abaqus, then sorts surface nodes into node sets and finally uses "*Equation" option in Abaqus to constrain the relationship between node sets and dummy nodes. The deformation of the RVE is prescribed by setting displacement values of the dummy nodes. The Python scripts used to implement PBCs are included in Appendix B. By changing the specific equations used to constrain the nodes, users can achieve either 2D or 3D PBC implementation for the RVE.

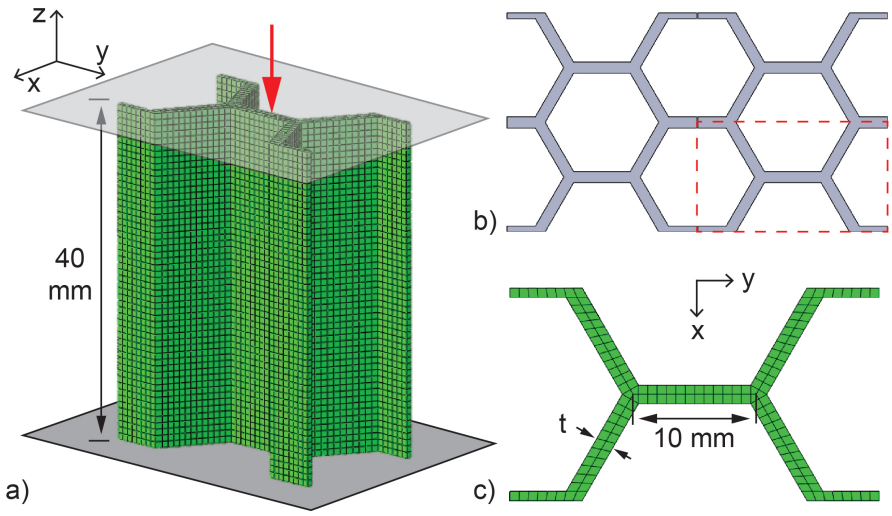

Figure 2.1: FEA model and selection of RVE. a) FEA model with mesh. The arrow shows the direction of compression by the rigid plate. b) Honeycomb array consisting of four RVEs. The red box cut out a RVE. c) Top view of the meshed RVE

2.2 FEA Setup

The honeycomb is meshed with eight-node brick elements (C3D8) in Abaqus. A rigid plate is placed at each end of the honeycomb. Uniaxial compression is applied by fixing the bottom plate while the top plate is given a constant velocity of 0.5 mm/s downward (in the negative z-direction, shown in Figure 2.1 a)). Hard contact normal behavior and tangential frictional behavior is defined as the surface interaction for both honeycomb self-contact and honeycomb-plate contact. To guide the buckling of the structure, an initial geometric imperfection is introduced by the "*Imperfection" option in Abaqus (Details see Appendix C). The first buckling mode obtained from elastic buckling analysis is used to shape the initial deformation. An elastic-plastic model with linear strain hardening typical of a printable polyurethane is used as the material, the strain-stress curve of which is shown in Figure 2.2. The elastic modulus and yield stress are 286 MPa

and 20 MPa respectively. The material fails at a strain of 2.73 and a stress of 29 MPa. The simulations are performed in Abaqus/Explicit. The final displacement of the rigid plate is 20 mm, resulting in an effective engineering strain of 0.5 for the honeycomb.

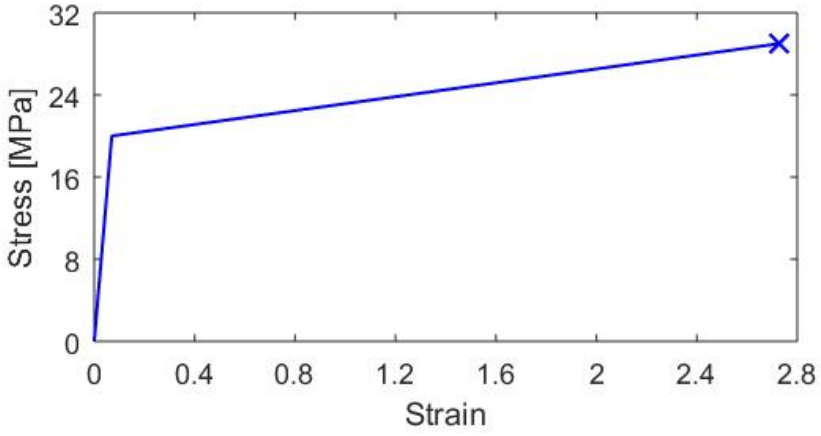

Figure 2.2: Material strain-stress curve. The cross at the end of the curve denotes the failure point

2.3 Sources of Modeling Uncertainty

Epistemic uncertainty in the finite element modeling comes from a variety of sources such as meshing and solver settings. For the buckling phenomena studied in this thesis, we consider the uncertainty introduced by the initial imperfection of the geometry and the tangential interaction (friction) between surfaces. These two factors directly affect the buckling process of the honeycomb and consequently lead to significantly different mechanical responses of the metamaterial. In Abaqus, the initial imperfection can be introduced by superposing multiple buckling modes of different magnitudes. Given that buckling is most likely to follow the pattern of the first mode, we only use the first buckling mode

to shape the initial deformation. Abaqus provides multiple options for a modeler to implement friction. For example, prescribing slip rate, contact pressure and temperature dependent data from user inputs, or specifying static and kinetic friction coefficients and decay rule, or including extra behaviors such as tangential softening and a shear stress limit. In this study, we use the default model where a constant friction coefficient is prescribed.

2.4 Choice of Simulations - Experimental design

Three mechanical properties are of interest for the design problem, namely effective stiffness, transition force, and specific energy absorption. They are described in more detail in Section 2.6. Response surfaces estimating total error with respect to target values of those parameters are built from the simulation results at selected sample points. Specifically, FEA is conducted on seven geometries with thickness (t) 1.0, 1.5, 2.0, 2.5, 3.0, 3.5 and 4.0 mm respectively. For each geometry, a 6×6 factorial design is used to characterize the effect of uncertainty in imperfection and friction – each taking six different values. An overview of the 6×6 experimental design is shown in Figure 2.3. The first buckling mode is imposed with magnitudes of 0.05, 0.1, 0.2, 0.3, 0.4 and 0.5 mm. Possible friction coefficients are 0, 0.01, 0.04, 0.1, 0.3 and 0.5. Each point in the figure is uniquely colored so that the corresponding results can be identified according to the color. Of note, typical simulations of buckling behavior use very small values for imperfections. For instance, Anbarasu [55] and Anapayan et al. [56] use one-thousands of the structure height as the imperfection magnitude. Friction is usually excluded from the analyses due to the lack of knowledge (e.g. [57, 58]). Given the purpose of characterizing how the uncertainty affects

the simulation results, a commonly adopted FEA setup should be used as the baseline case. Consequently, out of the 36 simulations for each geometry, the one with an imperfection magnitude of 0.1 mm and a friction coefficient of 0 is selected as the baseline case, shown in red circle in Figure 2.3.

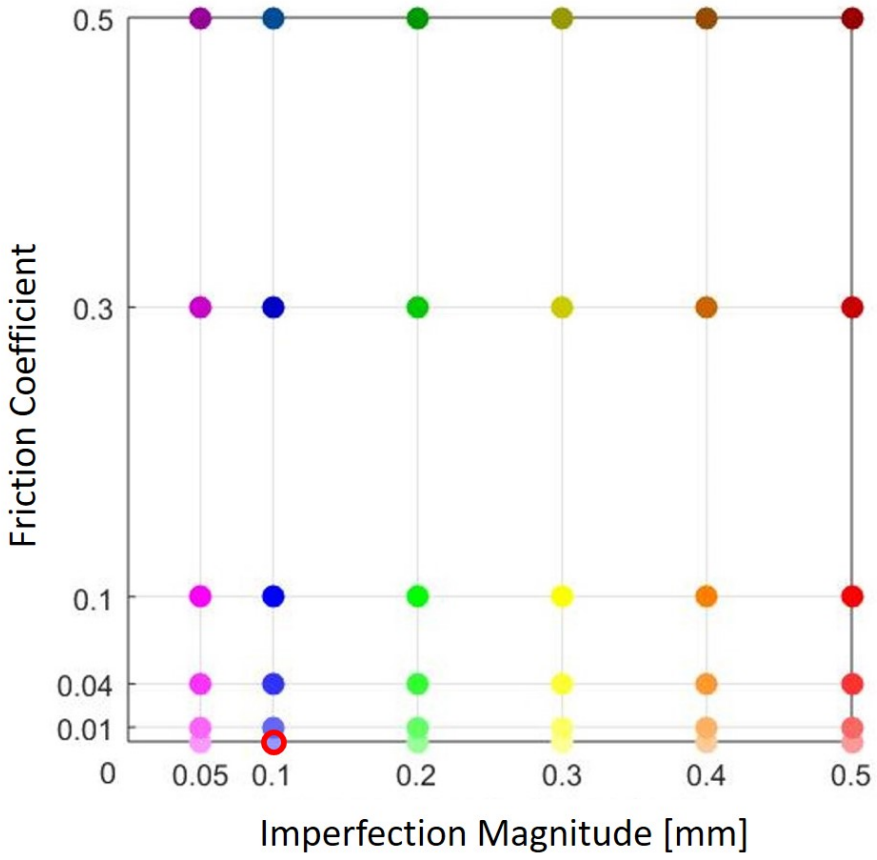

Figure 2.3: Choice of simulations in the 6×6 factorial design. Each point in the figure represents a simulation with a specific combination of imperfection and friction. The baseline case is circled in red. Points are uniquely colored so that the corresponding simulation results can be identified according to the color

2.5 Simulation Results

Figure 2.4 shows three deformed geometries (engineering strain = 0.5) that have minimum (t = 1.0 mm), median (t = 2.5 mm) and maximum (t = 4.0 mm) thicknesses respectively for the baseline case. It is clear that the thinnest honeycomb is subjected to severe buckling and extensive self-contact, while the thickest one shows very limited tendency towards buckling. The median thickness case presents a deformation pattern somewhere between the other two.

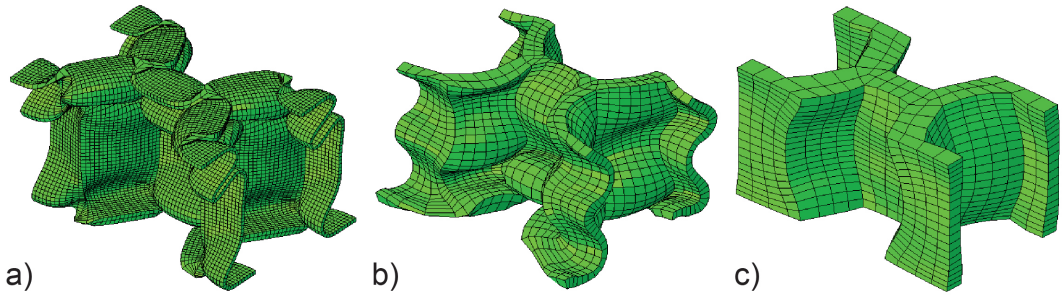

Figure 2.4: Three honeycomb geometries at the end of compression, i.e., at an engineering strain of 0.5 a) t = 1.0 mm b) t = 2.5 mm c) t = 4.0 mm

The direct observations of the deformed honeycomb agree with the results of reaction force response in Figure 2.5. In the figure, each bundle of curves comes from one value of thickness, as shown in the labels. The subplot embedded in the upper left corner is essentially Figure 2.3. The colors in the subplot are in one-to-one correspondence with those used for each bundle of curves. Due to excessive deformation and consequent element distortion, 5 of the 36 cases for the honeycomb with t = 1.0 mm were not successfully completed, and consequently, the analysis for this geometry is based on the 31 available simulation results. All 36 simulations are successfully completed for the other 6 geometries. For the two thinnest geometries, a sudden drop can be observed in some

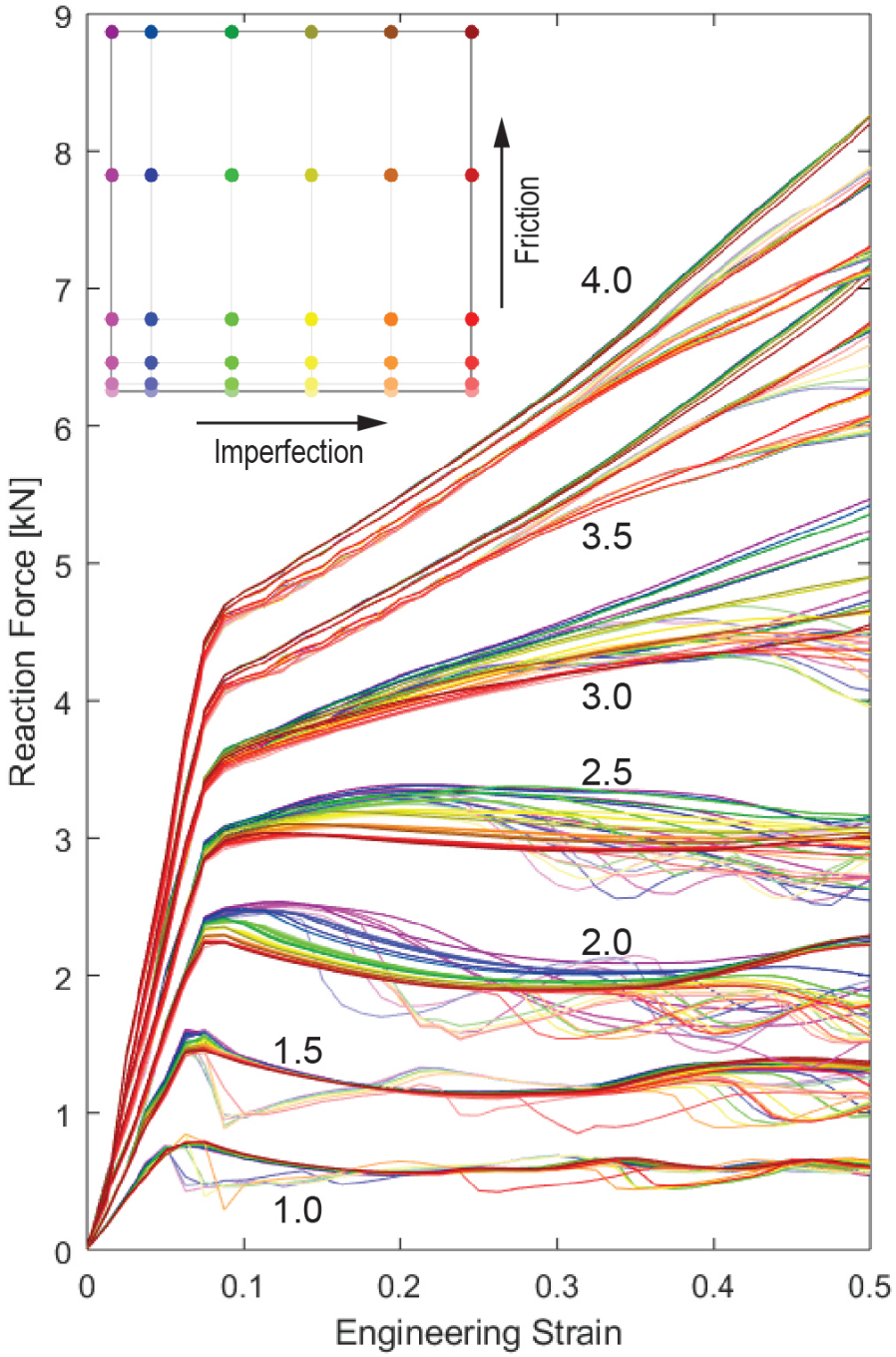

Figure 2.5: Force response of honeycombs during compression. Seven bundles of curves come from the corresponding seven geometries, the thicknesses of which are labeled beside. The subplot serves as the legend where the arrows indicate the increasing of imperfection magnitude and friction coefficient

curves after the reaction force reaches the maximum, which means a sudden loss of stability due to buckling. All these curves with force drops are in light colors, which correspond to cases with a low friction coefficient of either 0 or 0.01. Low friction makes the system prone to collapse, whereas high friction delays the process. For the two geometries with intermediate thickness, t = 2.0 and 2.5 mm, the force stays stable right after the turning point. The relatively small variability of different force responses is mainly caused by the imperfection as opposed to friction. However, variability is observed to increase substantially for larger values of strain. As for the thickest three geometries, they are relatively robust against the uncertainty introduced by imperfection and friction. The force response is stable through to moderate strain. After a strain of approximately 0.3, the curves start to diverge and this tendency exaggerates with the process of compression. Generally, the force responses for all seven geometries show considerable variability. Therefore, considering just one simulation with a set of imperfection and friction cannot capture the set of plausible non-linear mechanical responses one may observe in reality.

2.6 Performance Metrics and Effect of Uncertainties

The metrics for evaluating the mechanical behavior of the metamaterial can be extracted from the reaction force curve. Although a variety of metrics have been proposed to measure the traditional structural crashworthiness, many of them (e.g. peak force) cannot be directly adopted here because this study focuses on the design of printable polymer-based metamaterials for different applications, which extends the design domain to much larger wall-thickness-to-width ratio. The direct consequence is the different force progression pattern of the material

during compression. Consequently, other than the traditional SEA metric, we adopt an effective stiffness and linear-nonlinear transition force to measure the mechanical performance of the material. These are discussed in more detail below.

As observed in Figure 2.5, the metamaterial first shows a linear force-strain relationship. The extended design domain of wall thickness yields a much wider range in stiffness of the metamaterial compared to structural applications. Rather than the intrinsic material elastic modulus, the effective stiffness E_{eff} indicates the metamaterial's resistance to deformation under force in the linear region, defined as

$$E_{eff} = \frac{Fd}{Wh} \tag{2.1}$$

where F is the reaction force; d is the displacement of the rigid plate; W is the cross-sectional area (area of the box in Figure 2.1 b)); and h is the height of the RVE. After this linear region, the force either fluctuates severely or continues to progress stably at a different slope. The reaction force at the end of the linear region is therefore defined as transition force, TF. And finally, specific energy absorption is defined as:

$$SEA = \frac{Energy\ \ absorption}{Total\ \ mass} \tag{2.2}$$

where $Energy\ \ absorption$ is the area between the force-strain curve and horizontal axis in Figure 2.5. This widely used parameter describes the capability of absorbing external impact energy considering mass efficiency.

To identify how friction and imperfection separately change the mechanical performance of the material, i.e., the first-order sensitivities of these two factors,

TF, SEA and E_{eff} versus the two uncertainty sources are plotted in Figure 2.6, Figure 2.7 and Figure 2.8 respectively. In each figure, imperfection magnitude and friction coefficient share the same horizontal axis and their corresponding results are plotted in a blue solid line and a red dash-dot line respectively. Different markers are used to distinguish thickness, as annotated in the figure legend. Each data point on the solid curve is the average value of six cases (six different friction coefficients) at that given imperfection magnitude. And similarly, a point on the dash-dot curve shows the average value of all six cases with different imperfection magnitudes.

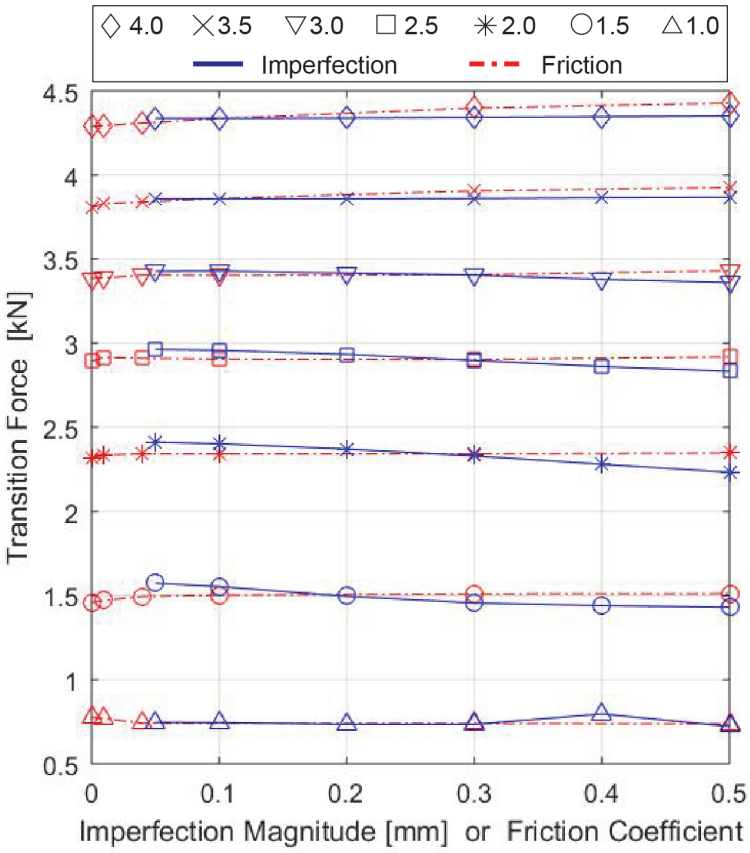

Figure 2.6: Effect of uncertainties on transition force. The effect due to imperfection magnitude is plotted in solid blue line and the effect brought by friction coefficient is in dash-dot red line. Different markers indicate different thickness values

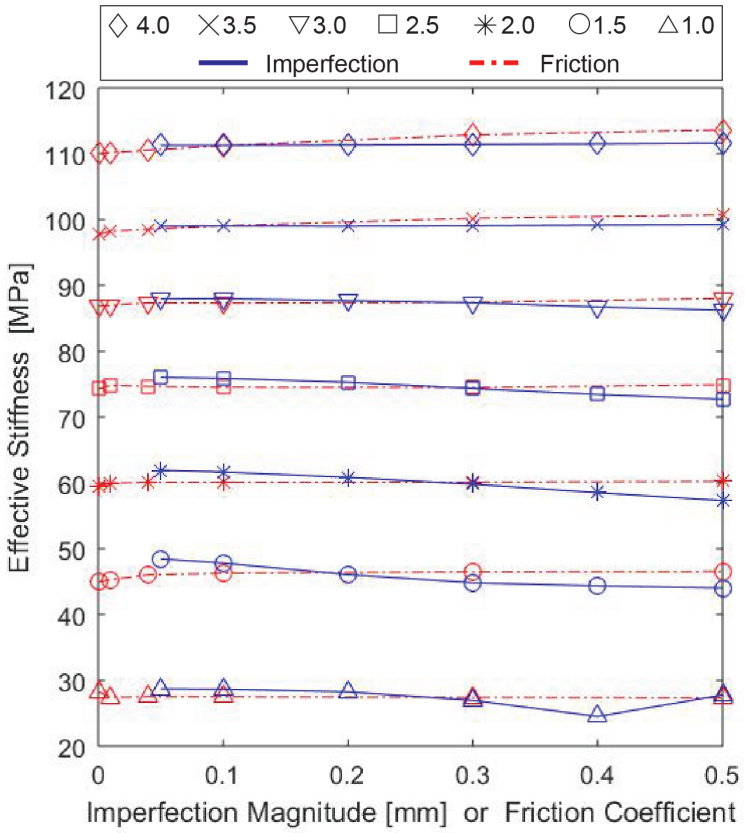

Figure 2.7: Effect of uncertainties on effective stiffness. The effect due to imperfection magnitude is plotted in solid blue line and the effect brought by friction coefficient is in dash-dot red line. Different markers indicate different thickness values

It can be seen that there is no general monotonic relationship between the mechanical responses and either friction or imperfection. SEA appears to be generally more sensitive to these two factors than the other metrics. From the perspective of geometry, the honeycombs with t = 1.5, 2.0, 2.5 mm are less robust against the uncertainties. The two thickest geometries (t = 3.5, 4.0 mm) are relatively stable to changes in imperfection but show significant performance change when different friction coefficients are applied. Table 2.6 reports the range ($max-min$) and standard deviation of each metric at each geometry. Quantitatively, these two uncertainty sources can bring a variation (range/mean) up

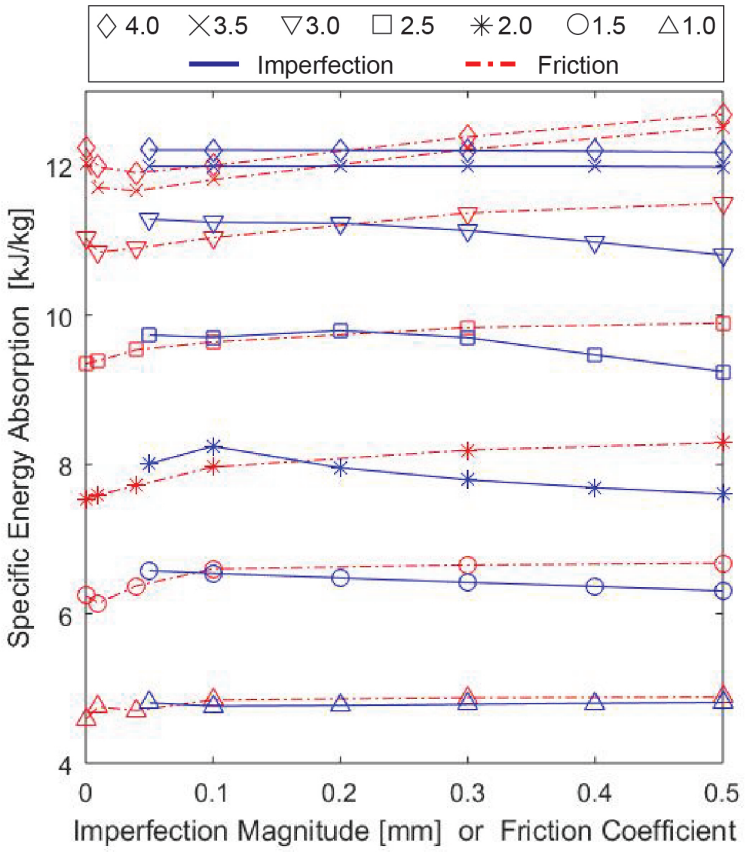

Figure 2.8: Effect of uncertainties on specific energy absorption. The effect due to imperfection magnitude is plotted in solid blue line and the effect brought by friction coefficient is in dash-dot red line. Different markers indicate different thickness values

to 20% for TF and E_{eff} at the thickness of 1.0 mm, and also up to 20% for SEA at the thickness of 2.0 mm.

According to Figures 2.6, 2.7 and 2.8, the friction coefficient generally has a stronger impact than imperfection magnitude on the performance of the honeycomb. Given that there is little tendency to relative motion between the honeycomb surfaces, this fact could result from the fact that friction between the rigid plate and the honeycomb changes the end constraint and therefore directly affects the buckling process. To verify this hypothesis, extra simulations

were conducted excluding the self-contact friction but still including the friction between honeycomb and rigid plates. Under this condition, the reaction force responses remained almost the same. This result suggests that it is indeed the friction between honeycomb and rigid plates that plays the major rule in affecting the buckling process.

Table 2.1: RANGE AND STANDARD DEVIATION OF EACH METRIC AT EACH GEOMETRY

t [mm]	TF [kN]		E_{eff} [MPa]		SEA [kJ/kg]	
	Range	Stdev	Range	Stdev	Range	Stdev
1.0	0.152	0.034	5.494	1.552	0.398	0.127
1.5	0.200	0.060	6.149	1.841	0.759	0.238
2.0	0.227	0.066	5.811	1.708	1.548	0.384
2.5	0.160	0.050	4.085	1.279	1.282	0.332
3.0	0.114	0.032	2.923	8.150	1.151	0.328
3.5	0.129	0.042	3.310	1.850	0.886	0.307
4.0	0.158	0.054	4.062	1.390	0.812	0.277

CHAPTER 3

ROBUST DESIGN APPLICATION

Chapter 2 showed how uncertainty in FEA simulations (from friction and imperfections) propagates onto mechanical properties of interest for a specific class of materials. This section shows the effect of those uncertainties in a design context. Specifically, a simple design problem is used to show the extent to which the modeling uncertainty can vary the design outcome. This simple problem provides insights that can be applicable for more complex design problems that are similar in physical essence but have higher dimensions and/or ill-behaved responses.

3.1 Problem Formulation

Following the notation in the robust compromise design framework [43], a deviation function (mean square error with respect to designer-specified target parameter values), denoted as $Z(t)$, is used as the objective function to be minimized with respect to the thickness value:

$$\begin{aligned} \min_{t} \quad & Z(t) \\ \text{s. t.} \quad & Z(t) = \left[\frac{1}{3}\left(\frac{E_{eff} - E_{eff\ target}}{E_{eff\ max}} \right)^2 \right. \\ & \qquad \left. + \frac{1}{3}\left(\frac{TF - TF_{target}}{TF_{max}} \right)^2 + \frac{1}{3}\left(\frac{SEA - SEA_{target}}{SEA_{max}} \right)^2 \right]^{\frac{1}{2}} \\ & 0 \le t \le t_{max} \end{aligned} \tag{3.1}$$

For the three metrics, subscript "*target*" denotes the design target and subscript

"*max*" means the maximum value out of all simulation results. Errors from E_{eff}, TF and SEA are normalized by their maximum values and weighted equally. Given a fixed wall width of 10 mm, the maximum thickness t_{max} is $5\sqrt{3}$ mm. In this problem, it is assumed that low transition force, medium stiffness and high specific energy absorption are preferred. Therefore, the target values of these three metrics are set as follows:

$$\left[\frac{TF_{target}}{TF_{max}} \quad \frac{E_{eff\ target}}{E_{eff\ max}} \quad \frac{SEA_{target}}{SEA_{max}}\right] = [0.3 \quad 0.6 \quad 1.0] \tag{3.2}$$

The maximum transition force, effective stiffness and specific energy absorption are 4.442 kN, 114.0 MPa and 12.71 kJ/kg respectively.

Note that the problem defined by Equation 3.1 is stochastic in nature due to the FEA uncertainty. In the context of design, and depending on the risk preferences of the decision maker, one may wish to minimize the worst-case combined error (minimax approach), or any target value of the cumulative distribution function of the error (e.g., median, mean + 1 stdev). We use a minimax approach in the following analysis. This particular problem instance is used in the remainder of the study to illustrate the main results. One thousand random instances of the same problem obtained from different combinations of target values were also used to assess the generality of the results.

3.2 Response Surface Method and Results

For each combination of model parameters in the 6×6 factorial design, the relationship between the deviation function $Z(t)$ and honeycomb wall thickness

t is described by a 5th order polynomial curve. The results are presented in Figure 3.1 (colors correlate to Figure 2.3). Each polynomial is a 1D response surface for one input (t) and one output (deviation Z). Alternatively, Figure 3.1 can be interpreted as a Gaussian process with a mean vector that follows a 5^{th} order (although roughly quadratic) polynomial and a certain covariance matrix. Figure 3.1 shows that the variability of $Z(t)$ is most significant at the thicknesses of 1.5 - 2.5 mm, indicating an area that is relatively less robust to uncertainty in imperfection and friction, while the error is smaller at both smaller and larger thicknesses. This is not expected because intuitively, the thinnest geometry goes through more severe unstable collapse and should be more sensitive to the change of outer condition. Figure 3.2 shows the statistics of the three metrics (TF, E_{eff}, SEA) at each design point (t). Here the error from target is directly defined as the normalized difference between the simulated property and the target property, e.g., $(TF - TF_{target})/TF_{max}$. It is apparent that SEA, corresponding to the error bars on the bottom curve, shows higher variability than either E_{eff} or TF across almost all thicknesses. This matches the results of the previous section. As evident in Figures 2.6 - 2.8, the mechanical responses are more spread around t = 2.0 mm, the thickness that is affected most by both imperfection and friction – close to the optimal thickness t^* in this problem. Compared with E_{eff} or TF, SEA experiences the highest variation with a change of either imperfection or friction.

The minimum for each curve in Figure 3.1 gives an optimum thickness under that certain combination of imperfection and friction. Figure 3.3 is a histogram of the optimal thicknesses. This thickness varies from 2.10 mm to 2.45 mm.

From the robust design perspective, adopting a worst-case strategy would

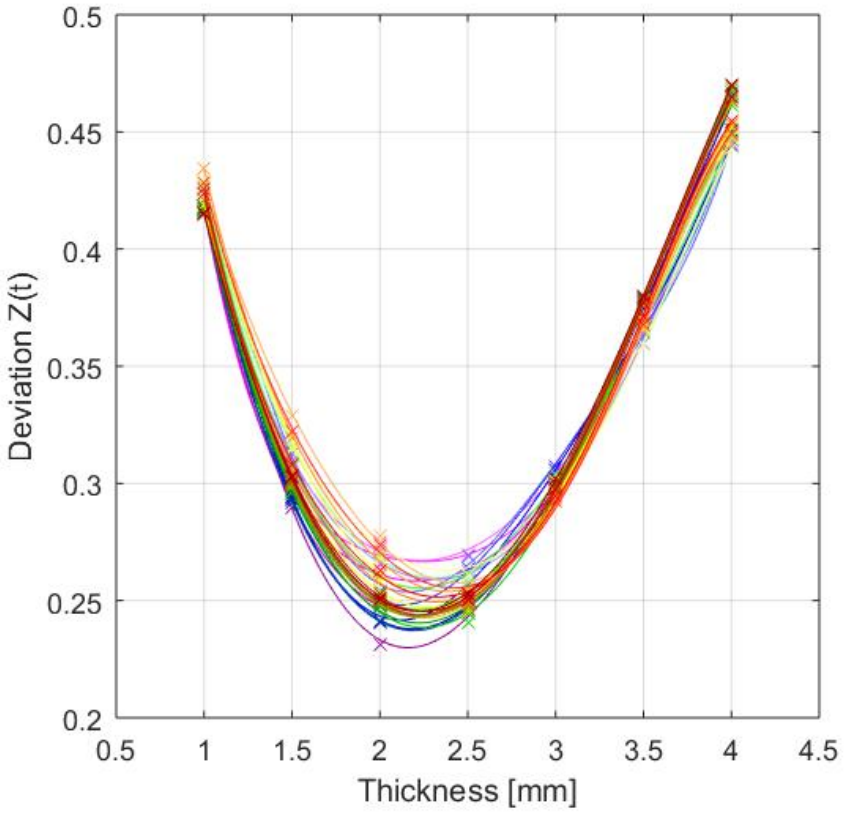

Figure 3.1: 1D response surfaces for deviation function-thickness relationship for all available cases. Curve colors match Figure 2.3

lead to selecting the value of thickness that leads to the smallest worst-case error (minimax approach) – in this case, 2.25 mm. Note that this thickness is substantially different from the optimal thickness for the baseline case, which is 2.32 mm, the 77.4th percentile of the optimum thickness distribution in Figure 3.3. From another point of view, using the optimal thickness for the baseline case, the actual error may be 0.266, instead of the optimal error of 0.235 predicted by the baseline case (36 additional simulations were run for t = 2.32 mm to compute this value.).

As noted in the introduction, many robust design methods make strong as-

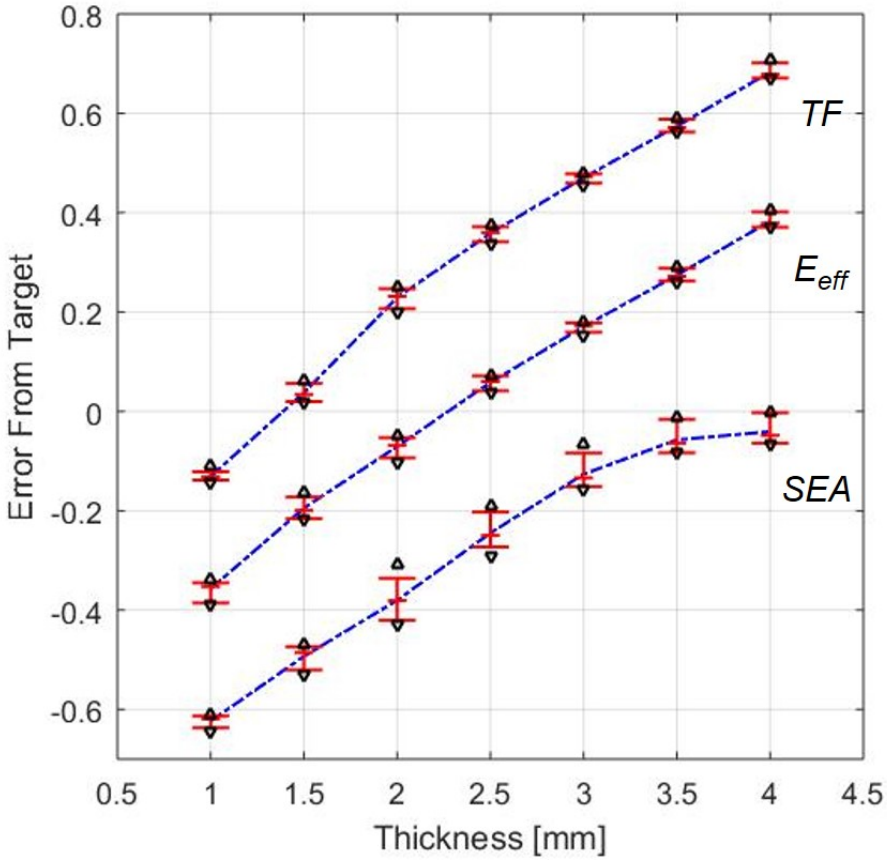

Figure 3.2: Statistics of the errors with respect to target values for each of the three metrics at each design point based on the 6×6 factorial design. The red horizontal bars show the 5th, 50th and 95th percentiles. The black triangles are the maximum and minimum values. The dashed blue curves connect the mean values

sumptions about the probability distribution of the uncertain parameters, often assuming some kind of normal distribution. Thus, a normality test should be run to determine whether the distribution of deviation function ($Z(t)$) due to FEA epistemic uncertainties is indeed normal. Henze-Zirkler's multivariate normality test, recommended by [59] because of its high power, is used here and we observe that this test rejects the hypothesis that the errors at all seven thicknesses are from a multivariate normal distribution with a p-value of nearly zero. This result casts some doubt on the use of normality assumption in a ro-

bust design framework. Univariate normality tests (with a significance level α = 0.05) are also conducted for the error distribution at each thickness, shown in Table 3.2. The Shapiro-Wilk test, recommended by some researchers as the best choice for testing univariate normality[60], does not reject the null hypothesis that the error is normally distributed at thickness t = 1.5 - 3.5 mm. Other major normality tests give similar judgments.

Table 3.1: UNIVARIATE NORMALITY TEST RESULTS AT EACH GEOMETRY. THE NULL HYPOTHESIS INDICATES THAT THE DATA IS NORMALLY DISTRIBUTED.

Normality Test	Reject null hypothesis or not (1 or 0)						
t :	1.0	1.5	2.0	2.5	3.0	3.5	4.0
Shapiro-Wilk	1	0	0	0	0	0	1
DAgostino-Pearson	0	0	0	0	0	0	1
Jarque-Bera	0	0	0	0	0	0	0
Lilliefors	1	0	0	0	1	1	1
Cramer-von Mises	1	0	0	1	0	0	1
Anderson-Darling	1	0	0	1	1	1	1

3.3 Results Generality

In order to assess the generality of the results above, 1000 design target vectors (as in Equation 3.2) were randomly generated with three target mechanical responses locating in the following intervals,

$$\frac{TF_{target}}{TF_{max}} \in [0.2 \;\; 1], \quad \frac{E_{eff\,target}}{E_{eff\,max}} \in [0.3 \;\; 1], \quad \frac{SEA_{target}}{SEA_{max}} \in [0.7 \;\; 1]$$

A higher SEA is desirable so the threshold was set as 0.7. Depending on specific applications, TF and E_{eff} can be selected from wider ranges. The lower

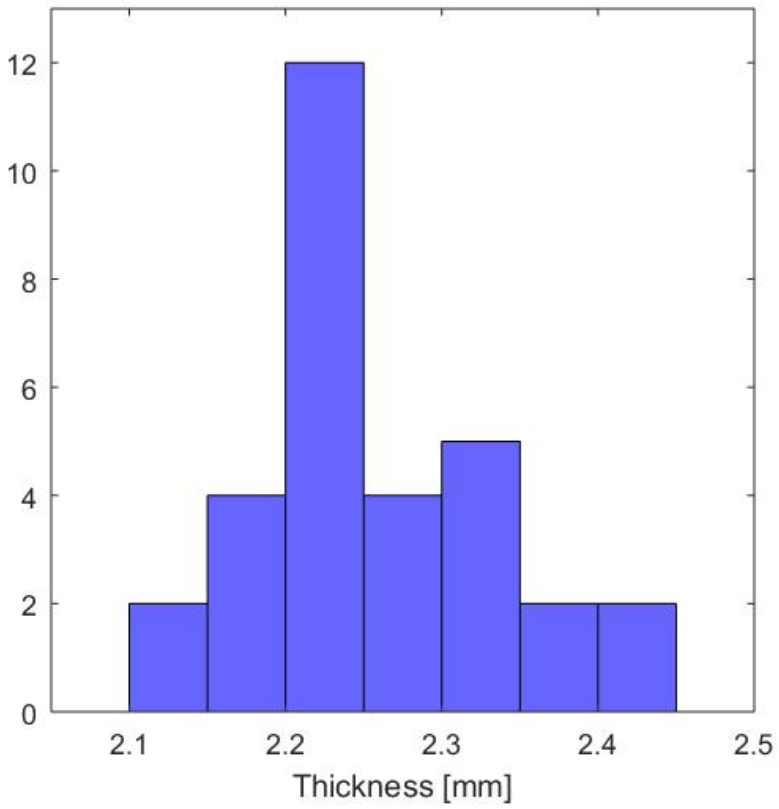

Figure 3.3: Histogram of optimum thickness from all uncertainty combinations

bounds were set to 0.2 and 0.3 respectively in order not to prescribe target values that lay outside the existing simulation outcomes, i.e., to make target values of each metric achievable. The results of these 1000 design problems are shown Figure 3.4. It includes two histograms showing both absolute and relative deviation in optimal thickness between the one prescribed using the baseline case and the one prescribed using a worst-case (minimax) approach accounting for the FEA uncertainty. The relative deviation is defined as follows:

$$\frac{t^*_{worstcase} - t^*_{baseline}}{t^*_{baseline}}$$

where t^* means the optimum thickness.

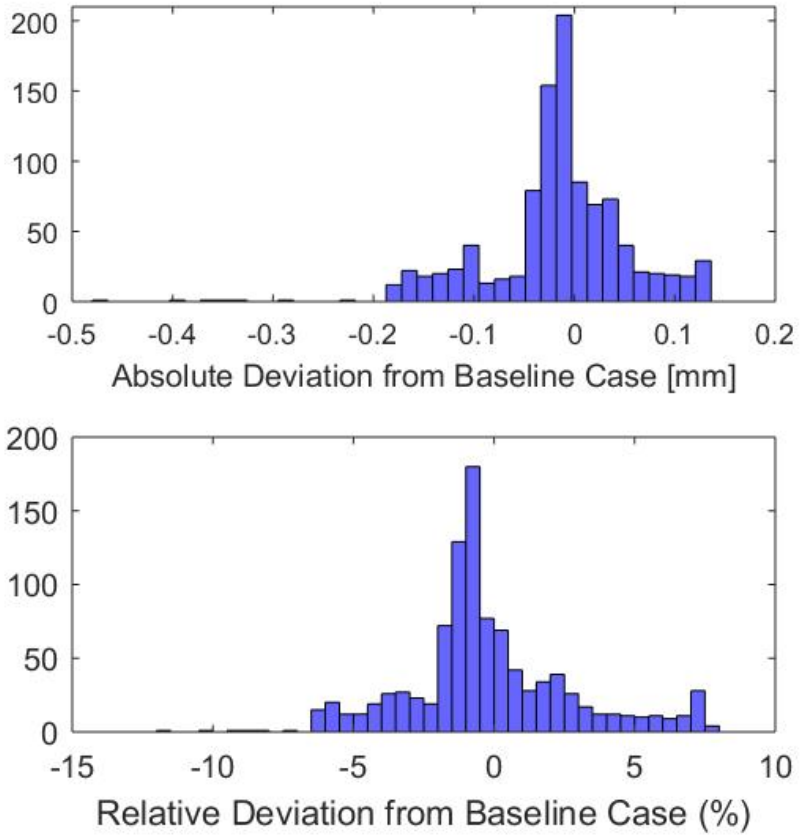

Figure 3.4: Histograms of deviation in optimal thickness between baseline case design and worst case design

We observe that neither distribution is centered around zero and some extreme cases bring a deviation of more than 0.4 mm or 10%. The mean of the absolute value of the deviation is 0.05 mm, or 2.18% (negative deviation values are replaced by their opposite numbers here). The histograms show that the baseline case can give a good approximation of a design problem for some scenarios, but fails in other situations.

3.4 Bayesian Optimization

Because of the expensive evaluations of the objective function, given the existing results, researchers would like to choose the next test/simulation point that has the highest potential to yield a better performance. As also noted in Chapter 1, Bayesian optimization is one of the favored tools to achieve the goal. Bayesian optimization utilizes machine learning techniques to infer the material properties based on the prior knowledge, which is the existing simulation outcomes in this study. For more details about its implementation and application in material design, please refer to [26]. Here a one-step Bayesian optimization is conducted to infer the current best solution to the problem (problem formulation 3.1), and the results are compared with those from the fitted polynomial response surface.

Bayesian optimization assumes the prior distribution of objective function $Z(t)$ to be normal:

$$\begin{bmatrix} Z(t) \\ Z(t') \end{bmatrix} \sim Normal \left(\begin{bmatrix} \mu(t) \\ \mu(t') \end{bmatrix}, \begin{bmatrix} cov(t,t) & cov(t,t') \\ cov(t',t) & cov(t',t') \end{bmatrix} \right) \tag{3.3}$$

Here t is generally a vector, denoting the thicknesses at which simulations have been performed. t' is the thicknesses whose mechanical performance is unknown and can be a scalar or a vector depending on how many points are to be predicted. μ and cov are the mean vector and variance-covariance matrix respectively. Usually the mean function can be set as zero or other constant, as a Gaussian process is flexible enough to model response without demanding precise prior knowledge of the mean function [61]. Here we set $\mu(t) = 0$ regardless

of the thickness. Here the Gaussian kernel (also known as radial basis function) is selected as the covariance function:

$$cov(Z(t_1), Z(t_2)) = m * exp(-\frac{1}{l^2}(t_1 - t_2)^2) \tag{3.4}$$

where t_1 and t_2 are any two geometry thicknesses. m and l are the so called hyper-parameters to determined. Under these assumptions, it can be proved that the predicted mean f and variance v at any new point t' are

$$f(t') = cov(t, t')^T * cov(t, t)^{-1} * Z(t) \tag{3.5}$$

$$v(t') = cov(t', t') - cov(t, t')^T * cov(t, t)^{-1} * cov(t, t') \tag{3.6}$$

In this way, we obtain the inferred objective value of the honeycomb at any thickness t' together with the confidence (variance) on the results. We can find the optimum thickness t^* that minimize the mean value as the optimal solution. However, this method only uses the information of mean value from Equation 3.5 and does not take the knowledge of distribution into consideration. The idea of expected improvement (EI) was proposed to balance the "exploitation" of the points with better mean values and the "exploration" of the points with high variances [62], defined as

$$EI(t) = E((Z^* - Z(t), 0)^+) \tag{3.7}$$

where Z^* is the current best solution of all existing evaluations and t is the point to be predicted. The notation $(x)^+$ means the maximum between x and zero. At

any new point t', the predicted value of $Z(t')$ follows a normal distribution. As the name suggests, EI indicates how much benefit we may get by conducting a simulation at the new point.

In this study, Bayesian optimization is first performed on the baseline case. The hyper-parameters m and l are determined by the method of maximizing log marginal likelihood:

$$\max_{m,l} \; log\Big(p(Z(t) \mid t)\Big) = -\frac{1}{2}\, Z(t)^T cov(t,t)^{-1} Z(t) \; - \; \frac{1}{2}\, log\Big(|cov(t,t)|\Big) + const. \qquad (3.8)$$

and it yields $m = 0.14$ and $l^2 = 1.5$. Figure 3.5 shows the results of this one-step problem, including the seven existing data points, predicted mean, 95% confidence interval (mean ± 1.96 stdev) and the expected improvement. It can be observed that the confidence interval is pretty narrow within the range we sampled (1.0 – 4.0 mm), which mean the objective is very well-behaved and the existing points yield a rather precise prediction of new points. The value of EI peaks at the thickness of 2.42 mm. Comparing this t^* obtained from maximizing EI to the results from the fitted polynomial response surface (2.32 mm), the difference is considerable.

Figure 3.6 shows the optimum thickness obtained by Bayesian optimization in all uncertainty combinations. Comparing to the results obtained by response surface method in Figure 3.3, the distribution is more uniform. Of note, no matter which method is adopted, the optimum thickness is spread in a range of 0.3 – 0.4 mm, which means quite a significant variability.

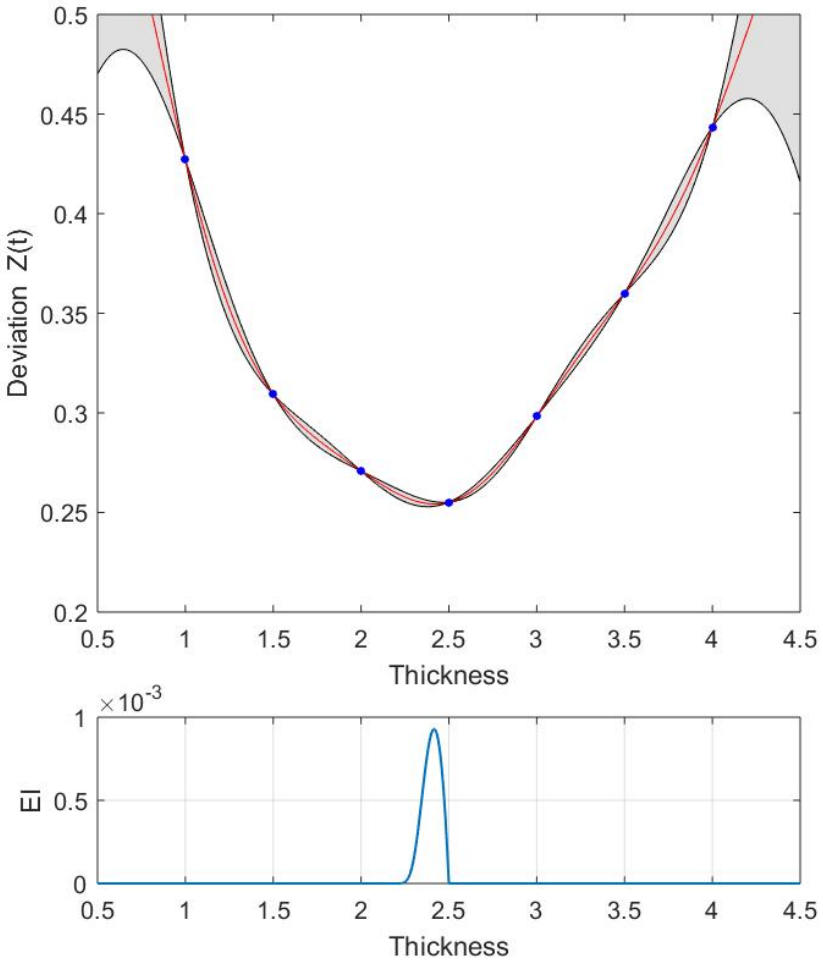

Figure 3.5: The blue dots are the original data points from the baseline case. Predicted mean values are plotted in a red curve. A 95% confidence interval is shown as the grey area, bounded by black curves. The expected improvement (EI) at each thickness is plotted below

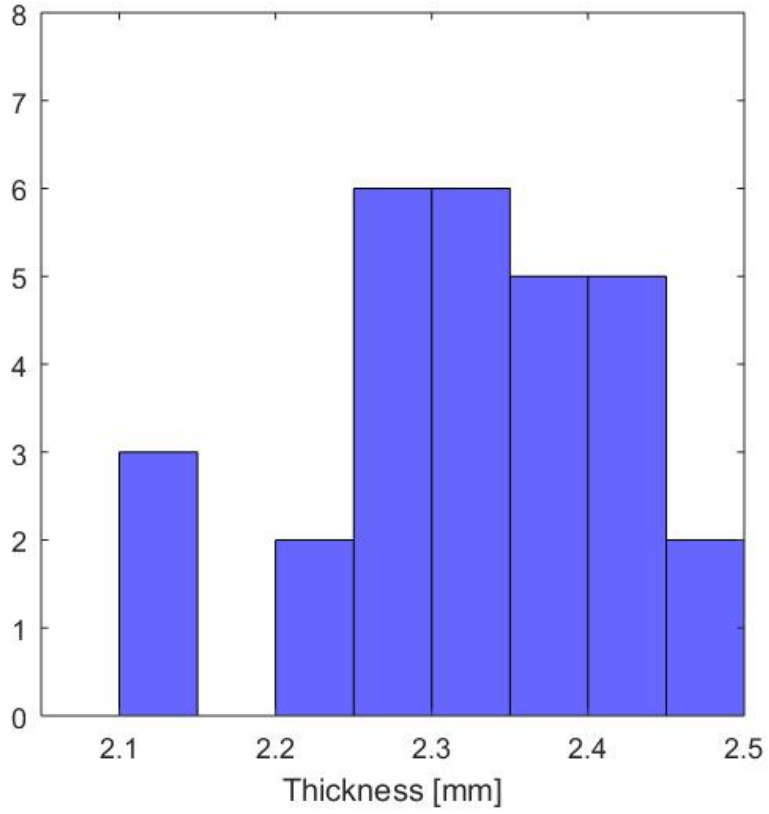

Figure 3.6: Histogram of the optimum thickness from Bayesian optimization for all uncertainty combinations

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

Previous research on design of cellular materials has largely ignored epistemic uncertainty in FEA simulations. This study has shown that, for a class of design problems with cellular materials working under large deformations, friction and imperfection parameter choices lead to variabilities of up to 20% in parameters of interest such as effective stiffness and specific energy absorption. When used in a context of a simple goal programming design framework, these uncertainties manifest themselves as deviations in the target values for parameters of interest for a nominally optimal thickness, or equivalently as changes in the optimal thickness values. Thus, failure to account for these uncertainties in FEA, such as friction and imperfections, can lead to significant errors in predictions of the mechanical response of the material during highly nonlinear behavior, and consequently alter the optimum designs. These results on a simple design problem for this class of materials suggest that uncertainties will be impactful for more complex design problems for which the complexity will make a priori anticipation of these impacts untenable. Results also show that the normality assumption behind several surrogate modeling strategies may be violated for this type of epistemic uncertainty, which suggests that designers should be careful when choosing surrogate models to substitute expensive FEA simulations.

In the future, more experiments can be performed to gather evidence of the importance of FEA epistemic uncertainty in robust design. The framework will be applied to a broader class of more complex problems with different materials and geometries. Other kinds of imperfections including ones that are more

realistic for printing will be analyzed. More generally, other sources of uncertainty such as meshing and general contact models will be considered. Different surrogate models and optimization algorithm will be used for comparison. The actual distribution, instead of just assuming normal distribution, can be incorporated into the model to achieve a possibly better effectiveness of optimization. Finally, models that predict FEA simulation failure could be used to avoid spending computational effort on simulations that are likely to fail.

APPENDIX A

PYTHON SCRIPTING FOR ABAQUS

Popular FEA packages incorporate a graphic user interface (GUI) and therefore enables users to build the model and process the results in a straightforward manner. However, if there are a great number of simulations with different parameters to run, it is cumbersome to do them one-by-one. Fortunately, operations in the Abaqus GUI can be manipulated by Python scripting so that the repetitive work can be coded to automatically build FEA models. Some basic knowledge is introduced in the article *Learn Abaqus Script in One Hour* [63]. A general idea is to perform operations in the GUI and check the corresponding syntax in the *.rpy* file that is automatically generated in the working directory by Abaqus. Running this file leads to the same results as operating in the GUI.

This appendix includes useful scripts the author wrote for the FEA automation in this study and they are applicable to other scenarios. The scripts for basic modeling are not included here since they can be obtained directly from the *.rpy* file. In the scripts, the codes are colored and formatted according to the convention of Python language. The comments are in green and explain the corresponding parts of the code.

A.1 Edit the Abaqus input file

An input file (*.inp*) can be generated after the modeling. It is basically a text file and contains all the information needed for running the simulation, such as mesh and boundary conditions. It is recommended to directly to modify

the parameters in the input file because it is straightforward and less prone to syntax errors. It also enables users to directly adjust the coordinates of certain nodes. Here is a sample script showing how to edit the friction coefficients and imperfection magnitudes in the input file using Python scripting.

```
import numpy as np # import library
fric=[0,0.01,0.04,0.1,0.3,0.5] #6 friction coefficients
mag=[0.05,0.1,0.2,0.3,0.4,0.5] #6 imperfection magnitudes
case=1
for i in mag:
      magstr=str(i/1000)
      for k in fric:
         fricstr=str(k)
         with open('case'+str(case)+'.inp','w') as file_2:
            file_1 = open('Job-NL.inp','r') # original input
               file
            case=case+1
            for line in file_1.readlines():
               # write identifiers in the .inp file in advance
               # so that it knows where to put the parameters
               # **mode1, **FricCoeff are identifiers
               if '**mode1' in line:
                  file_2.write('1,'+magstr+'\n')
               elif '**FricCoeff ' in line:
                  file_2.write(fricstr+','+'\n')
               else:
                  file_2.write(line) # other settings remain
                     the same
file_2.close()
file_1.close()
```

A.2 Submit batch jobs

Typical simulation jobs can be submitted directly in Abaqus/CAE but some complicated jobs are not allowed, e.g., the ones with the implementation of periodic boundary conditions in this study. These jobs must be submitted in the command prompt (Windows OS) or terminal (Linux OS) by input file. The following script shows how to generate commands for submitting jobs.

```
case=np.linspace(1,36,36,dtype=int).tolist()
#For Windows
file_3 = open('A_Run_Windows.bat', "w")
file_3.write('('+'\n')
for i in case:
    file_3.write('abaqus job=case'+str(i)+ '
        input=case'+str(i)+'.inp cpus=1 interactive' +'\n')#
file_3.write(')')
file_3.close()
#For Linux
file_3 = open('A_Run_Linux.bat', "w")
for i in case:
    file_3.write('abaqus job=case'+str(i)+ '
        input=case'+str(i)+'.inp cpus=1 interactive;')#
file_3.close()
```

In Windows OS, put the batch files in the same folder, otherwise the path should be specified. Executing the batch file will start the jobs sequentially. In Linux OS, however, things are more complicated. If the batch files are generated in DOS, they should first be converted to the unix format by the command *dos2unix*. Then you make it executable by *chmod +x*.

A.3 Export the field output to CSV

Users can indeed export the simulation results like field output to a text file via GUI but the format of the data is rigid and usually undesirable, which means extra work on post processing, especially for multiple cases. The following is a useful script to save the field output directly to a *.csv* file without directly interfacing with the Abaqus GUI.

```
#import libraries
from abaqus import *
from abaqusConstants import *
import odbAccess
import visualization
import numpy as np
```

```
import csv

Final_output= np.zeros([36,43],float) #initialize the variable
allcases=[1,...,36] #all cases in the 6 by 6 factorial design
foldername='t4.0' # the folder where the .odb files are

for k in allcases:
   #the name of the .odb
  string ='case'+ str(k)
  #path to the .odb
  job_path =
     '/home/computation/Desktop/'+foldername+'/'+string+'.odb'
  job_object = session.openOdb(name=job_path)

  #default operation in GUI. Not Necessary.
  session.viewports['Viewport:
     1'].setValues(displayedObject=job_object)

   #select the variable for output
   #here the reaction fore at 3 direction at the specific node
  session.xyDataListFromField(odb=job_object,
     outputPosition=NODAL, variable=(('RF',
    NODAL, ((COMPONENT, 'RF3'), )), ), nodeSets=(
    'REFERENCE_POINT_PART-5-1 9', ))

  #the following lines are not recorded in the .rpy file
  #convert the data to array so that it can be written to CSV
  RFdata_pt = session.xyDataObjects['RF:RF3 PI: PART-5-1 N: 9']
  RFdata_ptb= np.asarray(RFdata_pt,dtype=np.float64)
  Final_output[k-1,0:len(RFdata_ptb[:,1])]= RFdata_ptb[:,1]

  #delete and close the old file before open the next one
  #this prevent the confusion of naming
  del session.xyDataObjects['RF:RF3 PI: PART-5-1 N: 9']
  job_object.close()

# write the csv
with open('/home/computation/Desktop/'+foldername+'.csv','wb')
   as f:
  writer = csv.writer(f)
  writer.writerows(Final_output)

f.close()
```

APPENDIX B

IMPLEMENTATION OF PERIODIC BOUNDARY CONDITION

The following scripts shows how PBCs are configured in Abaqus. Note that the scripts are straightforward but not necessarily concise and efficient. It is however not a problem, given that the time running the script is negligible compared to the time spent on running simulations.

B.1 Read coordinates from input files

The coordinates of nodes can be obtained from the input file, so the first step is to read the coordinates from the text file.

```
import numpy as np

#get coordinates
f = open('Job-1.inp','r')
with open('all_coordinates.txt','w') as fc:
    # read a certain range of lines
    # the number may vary
    for line in f.readlines()[9:len(f.readlines())-11]:
        #read all nodes until the *Element appears
        if '*Element' in line:
            break
        else:
            fc.write(line)
fc.close()
f.close()

#read and convert to floats
fd = open('all_coordinates.txt','r')
data = np.loadtxt(fd,delimiter=',',dtype=(float))
fd.close()
fd = open('all_coordinates.txt','r')
data2 = np.loadtxt(fd, delimiter=',',dtype=(int))
fd.close()

node=data2[:,0]#node number
xcor=data[:,1]#x coordinates
ycor=data[:,2]#y coordinates
zcor=data[:,3]#z coordinates
```

B.2 Sort nodes into node sets

In this step, all surface nodes are sorted into node sets according to their coordinates. For a cube RVE, there are eight corners, twelve edges and six faces, and the nodes on each of them form a node set. The following code makes the corner set, and each set consists of just one node.

```
xmax=max(xcor) #get the maximum and minimum
xmin=min(xcor) #to identify the nodes on the surface
ymax=max(ycor)
ymin=min(ycor)
zmax=max(zcor)
zmin=min(zcor)

j=0;
for i in range(0,L):
   if j==8:break
   elif xcor[i] == xmax:
      if ycor[i]== ymax:
         if zcor[i]== zmax:
            # the node on the corner with maximum coordinates
            file.write('*Nset, nset=C1, instance=PART-1-1'+'\n'
                     + str(node[i])+'\n')
            j=j+1;
         elif zcor[i]== zmin:
               file.write('*Nset, nset=C2,
                  instance=PART-1-1'+'\n'
                        + str(node[i])+'\n')
               j=j+1;

      elif ycor[i]== ymin:
            if zcor[i]== zmax:
               file.write('*Nset, nset=C5,
                  instance=PART-1-1'+'\n'
                        + str(node[i])+'\n')
               j=j+1;
            elif zcor[i]== zmin:
                  file.write('*Nset, nset=C6,
                     instance=PART-1-1'+'\n'
                           + str(node[i])+'\n')
                  j=j+1;

   elif xcor[i]== xmin:
         if ycor[i]== ymax:
            if zcor[i]== zmax:
               file.write('*Nset, nset=C4,
                  instance=PART-1-1'+'\n'
                        + str(node[i])+'\n')
               j=j+1;
            elif zcor[i]== zmin:
```

```
40                    file.write('*Nset, nset=C3,
                         instance=PART-1-1'+'\n'
41                              + str(node[i])+'\n')
42                    j=j+1;
43
44            elif ycor[i]== ymin:
45                  if zcor[i]== zmax:
46                    file.write('*Nset, nset=C8,
                         instance=PART-1-1'+'\n'
47                              + str(node[i])+'\n')
48                    j=j+1;
49                  elif zcor[i]== zmin:
50                        file.write('*Nset, nset=C7,
                             instance=PART-1-1'+'\n'
51                                  + str(node[i])+'\n')
52                        j=j+1;
```

Each edge set generally contains more than one node and those already in the corner set should be excluded. Here four edges that are parallel to z axis are used as an example, shown as follows. Special attention should be paid to line 46, because each line in the input file should not contain more than 16 nodes, otherwise Abaqus reports an error. Also note that "unsorted" is add in the lines starting with *Nset. This command prevents Abaqus from reordering the nodes.

```
1  xy=np.zeros([int(L/12),4], float)
2  xy_=np.zeros([int(L/12),4], float)
3  x_y_=np.zeros([int(L/12),4], float)
4  x_y=np.zeros([int(L/12),4], float)
5
6  j=0;p=0;q=0;k=0;
7  for i in range(0,L):
8     if xcor[i]== xmax:
9        if ycor[i]== ymax:
10          #exclude the corner nodes
11          if zcor[i]!= zmax and zcor[i]!= zmin:
12             xy[j,0]=node[i];
13             xy[j,1]=xcor[i];
14             xy[j,2]=ycor[i];
15             xy[j,3]=zcor[i];
16             j=j+1
17       elif ycor[i]== ymin:
18          if zcor[i]!= zmax and zcor[i]!= zmin:
19             xy_[q,0]=node[i];
20             xy_[q,1]=xcor[i];
21             xy_[q,2]=ycor[i];
22             xy_[q,3]=zcor[i];
23             q=q+1;
```

```
    elif xcor[i]== xmin:
       if ycor[i]== ymax:
          if zcor[i]!= zmax and zcor[i]!= zmin:
             x_y[p,0]=node[i];
             x_y[p,1]=xcor[i];
             x_y[p,2]=ycor[i];
             x_y[p,3]=zcor[i];
             p=p+1;
       elif ycor[i]== ymin:
          if zcor[i]!= zmax and zcor[i]!= zmin:
             x_y_[k,0]=node[i];
             x_y_[k,1]=xcor[i];
             x_y_[k,2]=ycor[i];
             x_y_[k,3]=zcor[i];
             k=k+1;

xy2=xy[0:j,:] #eliminate redundant zeros
xy_2=xy_[0:q,:]
x_y2=x_y[0:p,:]
x_y_2=x_y_[0:k,:]

file.write('*Nset, nset=XY, instance=PART-1-1,unsorted'+'\n')
for n in range(0, j):
   if (n+1) % 16 == 0 and n != j-1: # no more than 16
      file.write(str(int(xy2[n,0])) + '\n')
   elif n != j-1:
      file.write(str(int(xy2[n,0])) + ',')
   else:
      file.write(str(int(xy2[n,0])) + '\n')
# sets X-Y XY- X-Y- are similar and thus omitted here
```

Finally, the face sets. They should exclude the nodes that already in corner and edge sets. The two faces vertical to x axis are used to demonstrate.

```
j=-1
p=-1
xpos=np.zeros([int(L/6),4], float)
xneg=np.zeros([int(L/6),4], float)

for i in range(0,L):
   if xcor[i]== xmax and ycor[i]!= ymax and ycor[i]!= ymin and
      zcor[i]!= zmax and zcor[i]!= zmin:
      j=j+1;
      xpos[j,0]=node[i];
      xpos[j,1]=xcor[i];
      xpos[j,2]=ycor[i];
      xpos[j,3]=zcor[i];
   elif xcor[i]== xmin and ycor[i]!= ymax and ycor[i]!= ymin
      and zcor[i]!= zmax and zcor[i]!= zmin:
      p=p+1;
      xneg[p,0]=node[i];
      xneg[p,1]=xcor[i];
```

```
      xneg[p,2]=ycor[i];
      xneg[p,3]=zcor[i];

xpos2=xpos[0:j+1,:] #eliminate redundant zeros
xneg2=xneg[0:p+1,:]

tt=np.vstack((xpos2,xneg2))

node2=tt[:,0]
xcor2=tt[:,1]
ycor2=tt[:,2]
zcor2=tt[:,3]
l=len(xcor2)
neg_sequence=np.copy(node2)

for k in range (0,int(l/2)):
   for i in range (int(l/2), l):
      if ycor2[i]==ycor2[k] and zcor2[i]==zcor2[k]:
            neg_sequence[k]=node2[i]

file.write('*Nset, nset=X_POS,
   instance=PART-1-1,unsorted'+'\n')
for n in range(0, int(l/2)):
   if (n+1) % 16 == 0 and n != int(l/2)-1:
      file.write(str(int(node2[n])) + '\n')
   elif n != int(l/2)-1:
      file.write(str(int(node2[n])) + ',')
   else:
      file.write(str(int(node2[n])) + '\n')
file.write('*Nset, nset=X_NEG,
   instance=PART-1-1,unsorted'+'\n')
for n in range(0, int(l/2)):
   if (n+1) % 16 == 0 and n != int(l/2)-1:
      file.write(str(int(neg_sequence[n])) + '\n')
   elif n != int(l/2)-1:
      file.write(str(int(neg_sequence[n])) + ',')
   else:
      file.write(str(int(neg_sequence[n])) + '\n')
```

B.3 Add constraints by *Equation

Wu [54] already clearly shows details of adding constraints to RVEs by the *Equation option in Abaqus. Consequently it is not further explained here. However, it is noteworthy that the number of equations depends the specific type of element used. For instance, if brick element is used to discretize the ge-

ometry, then all three translation degrees of freedom (DoF) of each node should be constrained. Alternatively, if shell element is applied, then the rotation DoFs should also be constrained. Changing the number of equations used can achieve either 2D or 3D PBC configuration for the RVE.

B.4 Robust PBC

It can be seen that the above scripts for implementing PBC pairs the nodes whose coordinate values are matched exactly. Unfortunately, this method does not always work. For some complicated geometries with unstructured meshes like tetrahedron, the coordinates of nodes generally do not match each other. Even for some simple geometry, the method is not robust enough because it does not allow any node location mismatch in the meshing process. To solve the problem, the pairing tolerance is added to achieve the robust implementation of PBC. See the following example for two face sets.

```
j=-1
p=-1
xpos=np.zeros([int(L/6),4], float)
xneg=np.zeros([int(L/6),4], float)

for i in range(0,L):
   #tol is a pre-determined value by users
   #it depends on geometry scale and mesh size
   if abs(xcor[i] - xmax) <= tol and abs(ycor[i]- ymax)>= tol
      and abs(ycor[i]- ymin)>= tol and abs(zcor[i]- zmax)>=
      tol and abs(zcor[i]- zmin)>= tol:
      j=j+1;
      xpos[j,0]=node[i];
      xpos[j,1]=xcor[i];
      xpos[j,2]=ycor[i];
      xpos[j,3]=zcor[i];
   elif abs(xcor[i] - xmin) <= tol and abs(ycor[i]- ymax)>=
      tol and abs(ycor[i]- ymin)>= tol and abs(zcor[i]-
      zmax)>= tol and abs(zcor[i]- zmin)>= tol:
      p=p+1;
      xneg[p,0]=node[i];
```

```
18          xneg[p,1]=xcor[i];
19          xneg[p,2]=ycor[i];
20          xneg[p,3]=zcor[i];
21
22  xpos=xpos[0:j+1,:]
23  xneg=xneg[0:p+1,:]
24
25  neg_sequence = [0]*(j+1)
26  for k in range (0,j+1):
27     distance1=distance0
28     #select one face as reference
29     #pair the nearest nodes on the other face
30     for i in range (0,p+1):
31        distance2=np.sqrt(np.square(xneg[i,2] - xpos[k,2]) +
              np.square(xneg[i,3] - xpos[k,3]))
32        if distance2 <= distance1:
33          distance1=distance2
34          neg_sequence[k]=int(xneg[i,0])
```

Note that a node may be paired to multiple nodes on the opposite face but this is generally not an issue if the mesh is fine enough. Again, adding "unsorted" is necessary because it allows repetitive node numbers in the same node set. The robust PBC has been successfully tested on a double-gyroid RVE.

APPENDIX C

APPLY IMPERFECTIONS IN ABAQUS

Applying imperfections in Abaqus can be achieved by superposing buckling modes to the geometry with specified magnitudes. For the linear buckling analysis, Abaqus/Standard performs eigenvalue buckling estimation using the keyword "*Buckle". See the following lines in the input file:

```
*Step, name=Step-1, nlgeom=NO, perturbation
*Buckle
n1 ,n2, n3, n4
```

Here n1 ,n2, n3 and n4 are all integers. n1 is the number of eigenvalues to be estimated, i.e., number of modes desired. n2 is the maximum eigenvalue of interest and it can be left blank. n3 is the number of vectors used in the iteration. n4 is the maximum number of iterations.

Again, the these lines can be obtained by generating input files after operations in the GUI. However, the following lines must be directly written to the input file before "*End Step":

```
*Output, history, variable=PRESELECT
*NODE FIL
U,
```

These three lines request Abaqus to output the nodal displacement after analysis. Multiple files will be generated in the working directory after the simulation job is done and they contain different information. The *.fil* and *.prt* are the two files required to apply imperfections. They should be placed in the folder where the final buckling simulation (in Abaqus/Explicit) is performed.

In the input file for the explicit analysis, imperfections are superposed by the

following commands:

```
*Imperfection, FILE=Job-Linear, STEP=1
mode1, magnitude1
mode2, magnitude2
mode3, magnitude3
*Step, name=Step-1, nlgeom=YES
```

The first line is the "*Imperfection" keyword mentioned in Section 2.2. File name is specified as "Job-Linear" just as an example, which means that Abaqus will look for the two files "Job-Linear.prt" and "Job-Linear.fil" to get the buckling modes. Here mode numbers are integers and the corresponding magnitudes are decimal numbers. The choice of modes and magnitudes are up to the users but usually lower order modes (modes with larger eigenvalues) are used. Note the imperfection should be prescribed before "*Step".

BIBLIOGRAPHY

[1] Wang, L., Boyce, M. C., Wen, C.-Y., and Thomas, E. L., 2009. "Plastic dissipation mechanisms in periodic microframe-structured polymers". *Advanced Functional Materials,* **19**(9), pp. 1343–1350.

[2] Wang, L., and Boyce, M. C., 2010. "Bioinspired structural material exhibiting post-yield lateral expansion and volumetric energy dissipation during tension". *Advanced Functional Materials,* **20**(18), pp. 3025–3030.

[3] Schaedler, T. A., and Carter, W. B., 2016. "Architected cellular materials". *Annual Review of Materials Research,* **46**, pp. 187–210.

[4] Bertoldi, K., Vitelli, V., Christensen, J., and van Hecke, M., 2017. "Flexible mechanical metamaterials". *Nature Reviews Materials*, **2**(11), p. 17066.

[5] Christensen, J., Kadic, M., Kraft, O., and Wegener, M., 2015. "Vibrant times for mechanical metamaterials". *MRS Communications,* **5**(3), pp. 453–462.

[6] Reddy, J. N., 2014. *An Introduction to Nonlinear Finite Element Analysis: with applications to heat transfer, fluid mechanics, and solid mechanics*. OUP Oxford.

[7] Fang, J., Sun, G., Qiu, N., Kim, N. H., and Li, Q., 2017. "On design optimization for structural crashworthiness and its state of the art". *Structural and Multidisciplinary Optimization,* **55**(3), pp. 1091–1119.

[8] Wierzbicki, T., 1983. "Crushing analysis of metal honeycombs". *International Journal of Impact Engineering*, **1**(2), pp. 157–174.

[9] Abramowicz, W., and Wierzbicki, T., 1989. "Axial crushing of multicorner sheet metal columns". *Journal of Applied Mechanics,* **56**(1), pp. 113–120.

[10] Paik, J. K., Thayamballi, A. K., and Kim, G. S., 1999. "The strength characteristics of aluminum honeycomb sandwich panels". *Thin-walled structures,* **35**(3), pp. 205–231.

[11] Yasui, Y., 2000. "Dynamic axial crushing of multi-layer honeycomb panels and impact tensile behavior of the component members". *International Journal of Impact Engineering*, **24**(6-7), pp. 659–671.

[12] Wu, E., and Jiang, W.-S., 1997. "Axial crush of metallic honeycombs". *International Journal of Impact Engineering,* **19**(5-6), pp. 439–456.

[13] Heimbs, S., Schmeer, S., Middendorf, P., and Maier, M., 2007. "Strain rate effects in phenolic composites and phenolic-impregnated honeycomb structures". *Composites Science and Technology,* **67**(13), pp. 2827–2837.

[14] Yin, H., Wen, G., Hou, S., and Chen, K., 2011. "Crushing analysis and multiobjective crashworthiness optimization of honeycomb-filled single and bitubular polygonal tubes". *Materials & Design,* **32**(8-9), pp. 4449–4460.

[15] Yin, H., Wen, G., and Gan, N., 2011. "Crashworthiness design for honeycomb structures under axial dynamic loading". *International Journal of Computational Methods,* **8**(04), pp. 863–877.

[16] Sun, G., Li, G., Stone, M., and Li, Q., 2010. "A two-stage multi-fidelity optimization procedure for honeycomb-type cellular materials". *Computational Materials Science,* **49**(3), pp. 500–511.

[17] Smit, R., Brekelmans, W., and Meijer, H., 1999. "Prediction of the large-strain mechanical response of heterogeneous polymer systems: local and global deformation behaviour of a representative volume element of voided polycarbonate". *Journal of the Mechanics and Physics of Solids,* **47**(2), pp. 201 – 221.

[18] Danielsson, M., Parks, D., and Boyce, M., 2002. "Three-dimensional micromechanical modeling of voided polymeric materials". *Journal of the Mechanics and Physics of Solids,* **50**(2), pp. 351 – 379.

[19] Matthews, J., Klatt, T., Seepersad, C. C., Haberman, M., and Shahan, D., 2013. "Hierarchical Design of Composite Materials with Negative Stiffness Inclusions Using a Bayesian Network Classifier". In ASME Des. Eng. Tech. Conf., pp. 1–13.

[20] Simpson, T. W., and Poplinski, J. D., 2001. "Metamodels for computer-based engineering design: survey and recommendations". *Eng. Comput.,* **17**, pp. 129–150.

[21] Montgomery, D. C., 2010. *Design and Analysis of Experiments*, 8 ed. Wiley, New York.

[22] Wang, S. Y., and Wang, M. Y., 2006. "Radial basis functions and level set method for structural topology optimization". *Int. J. Numer. Methods Eng.,* **65**(12), pp. 2060–2090.

[23] Yildiz, A. R., Öztürk, N., Kaya, N., and Öztürk, F., 2003. "Integrated optimal topology design and shape optimization using neural networks". *Struct. Multidiscip. Optim.,* **25**(4), pp. 251–260.

[24] Sanz-Corretge, J., 2017. "A procedure to design optimum composite plates using implicit decision trees". *Struct. Multidiscip. Optim.,* **56**(5), pp. 1169–1183.

[25] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., 1989. "Design and Analysis of Computer Experiments". *Stat. Sci.,* **4**(4), pp. 409–423.

[26] Frazier, P. I., and Wang, J., 2016. "Bayesian optimization for materials design". In *Information Science for Materials Discovery and Design*. Springer, pp. 45–75.

[27] Radhakrishnan, R., and McAdams, D. a., 2005. "A Methodology for Model Selection in Engineering Design". *J. Mech. Des.,* **127**(May), p. 378.

[28] Forrester, A. I. J., Sóbester, A., and Keane, A. J., 2007. "Multi-Fidelity Optimization via Surrogate Modelling". *Proc. Math. Phys. Eng. Sci.,* **463**(2088), pp. pp. 3251—-3269.

[29] Ghoreishi, S. F., and Allaire, D. L., 2018. "A Fusion-Based Multi-Information Source Optimization Approach using Knowledge Gradient Policies". *2018 AIAA/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf.*(January), pp. 1–14.

[30] Zhang, J., Chowdhury, S., Mehmani, A., and Messac, A., 2014. "Characterizing Uncertainty Attributable to Surrogate Models". *J. Mech. Des.,* **136**(3), p. 031004.

[31] Aughenbaugh, J. M., and Paredis, C. J. J., 2006. "The Value of Using Imprecise Probabilities in Engineering Design". *J. Mech. Des.,* **128**(July), p. 969.

[32] Ghosh, D. D., and Olewnik, A., 2013. "Computationally Efficient Imprecise Uncertainty Propagation". *J. Mech. Des.,* **135**(May), p. 51002.

[33] Thurston, D. L., and Carnahan, J. V., 1992. "Fuzzy Ratings and Utility Analysis in Preliminary Design Evaluation of Multiple Attributes". *J. Mech. Des.,* **114**(December 1992), pp. 648–658.

[34] Vanegas, L. V., and Labib, A. W., 2005. "Fuzzy Approaches to Evaluation in Engineering Design". *J. Mech. Des.,* **127**(January), p. 24.

[35] Agarwal, H., Renaud, J. E., Preston, E. L., and Padmanabhan, D., 2004. "Uncertainty quantification using evidence theory in multidisciplinary design optimization". *Reliab. Eng. Syst. Saf.,* **85**(1-3), pp. 281–294.

[36] Rao, S. S., and Annamdas, K. K., 2013. "A Comparative Study of Evidence Theories in the Modeling, Analysis, and Design of Engineering Systems". *J. Mech. Des.,* **135**(6), p. 61006.

[37] Taguchi, G., Elsayed, E. A., and Hsiang, T. C., 1989. *Quality engineering in production systems*. McGraw-Hill, New York.

[38] Allen, J. K., Seepersad, C., Choi, H., and Mistree, F., 2006. "Robust Design for Multiscale and Multidisciplinary Applications". *Journal of Mechanical Design,* **128**(4), p. 832.

[39] Schumacher, A., and Olschinka, C., 2008. "Robust design considering highly nonlinear structural behavior". *Structural and Multidisciplinary Optimization,* **35**(3), pp. 263–272.

[40] Neufville, R. D., Asce, L. M., Scholtes, S., and Wang, T., 2006. "Real Options by Spreadsheet : Parking Garage Case Example Valuing Real Options by Spreadsheet : Parking Garage Case Example". *Journal of Infrastructure Systems,* **12**(2), pp. 1–19.

[41] Smaling, R., and De Weck, O., 2007. "Assessing risks and opportunities of technology infusion in system design". *Systems Engineering,* **10**(1), pp. 1–25.

[42] Hu, Y., and Rao, S. S., 2011. "Robust Design of Horizontal Axis Wind Turbines Using Taguchi Method". *Journal of Mechanical Design,* **133**(11), p. 111009.

[43] Chen, W., Allen, J. K., Tsui, K.-L., and Mistree, F., 1996. "A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors". *Journal of Mechanical Design,* **118**(December), pp. 478–485.

[44] Ignizio, J. P., 1983. "Generalized goal programming - An overview". *Computers & Operations Research,* **10**(4), pp. 277–289.

[45] Mistree, F., Hughes, O. F., and Bras, B., 1993. "The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm". *Structural Optimization: Status and Promise*(1993), pp. 247–286.

[46] Beyer, H. G., and Sendhoff, B., 2007. "Robust optimization - A comprehensive survey". *Computer Methods in Applied Mechanics and Engineering*, **196**(33-34), pp. 3190–3218.

[47] Park, G.-J., Lee, T.-H., Lee, K. H., and Hwang, K.-H., 2006. "Robust Design: An Overview". *AIAA Journal*, **44**(1), pp. 181–191.

[48] Zhang, Y., and Hosder, S., 2013. "Robust Design Optimization Under Mixed Uncertainties With Stochastic Expansions". *Journal of Mechanical Design*, **135**(8), p. 081005.

[49] Suh, N. P., 1998. "Axiomatic design theory for systems". *Research in Engineering Design*, **10**(4), pp. 189–209.

[50] Lu, X., and Li, H.-X., 2011. "Robust Design for Dynamic System Under Model Uncertainty". *Journal of Mechanical Design*, **133**(2), p. 021006.

[51] Markowitz, H., 1952. "Portfolio selection". *The Journal of Finance*, **7**(1), pp. 77–91.

[52] Shah, P., Davendralingam, N., and DeLaurentis, D. A., 2015. "A conditional value-at-risk approach to risk management in system-of-systems architectures". In System of Systems Engineering Conference (SoSE), 2015 10th, IEEE, pp. 457–462.

[53] Yang, C., and Du, X., 2014. "Robust Design for Multivariate Quality Characteristics Using Extreme Value Distribution". *Journal of Mechanical Design*, **136**(10), p. 101405.

[54] Wu, W., Owino, J., Al-Ostaz, A., and Cai, L., 2014. "Applying periodic boundary conditions in finite element analysis". In SIMULIA Community Conference, Providence, pp. 707–719.

[55] Anbarasu, M., 2016. "Local-distortional buckling interaction on cold-formed steel lipped channel beams". *Thin-Walled Structures*, **98**(January), pp. 351–359.

[56] Anapayan, T., and Mahendran, M., 2012. "Numerical modelling and design of LiteSteel Beams subject to lateral buckling". *Journal of Constructional Steel Research,* **70**(March), pp. 51–64.

[57] Yin, H., Wen, G., and Gan, N., 2011. "Crashworthiness design for honeycomb structures under axial dynamic loading". *International Journal of Computational Methods,* **8**(4), pp. 863–877.

[58] Sun, G., Li, G., Stone, M., and Li, Q., 2010. "A two-stage multi-fidelity optimization procedure for honeycomb-type cellular materials". *Computational Materials Science,* **49**(3), pp. 500–511.

[59] Thode, H., 2002. *Testing for Normality New York*. Marcel Dekker Inc.

[60] Ghasemi, A., and Zahediasl, S., 2012. "Normality tests for statistical analysis: a guide for non-statisticians". *International journal of endocrinology and metabolism,* **10**(2), p. 486.

[61] Robert, C., 2014. *Machine learning, a probabilistic perspective*. Taylor & Francis.

[62] Mockus, J., 2012. *Bayesian approach to global optimization: theory and applications*, Vol. 37. Springer Science & Business Media.

[63] Overvelde, J., 2010. "Learn abaqus script in one hour". *http://www.overvelde.com/wpcontent/uploads/2011/05/LearnAbaqusScriptInOneHour. pdf*.